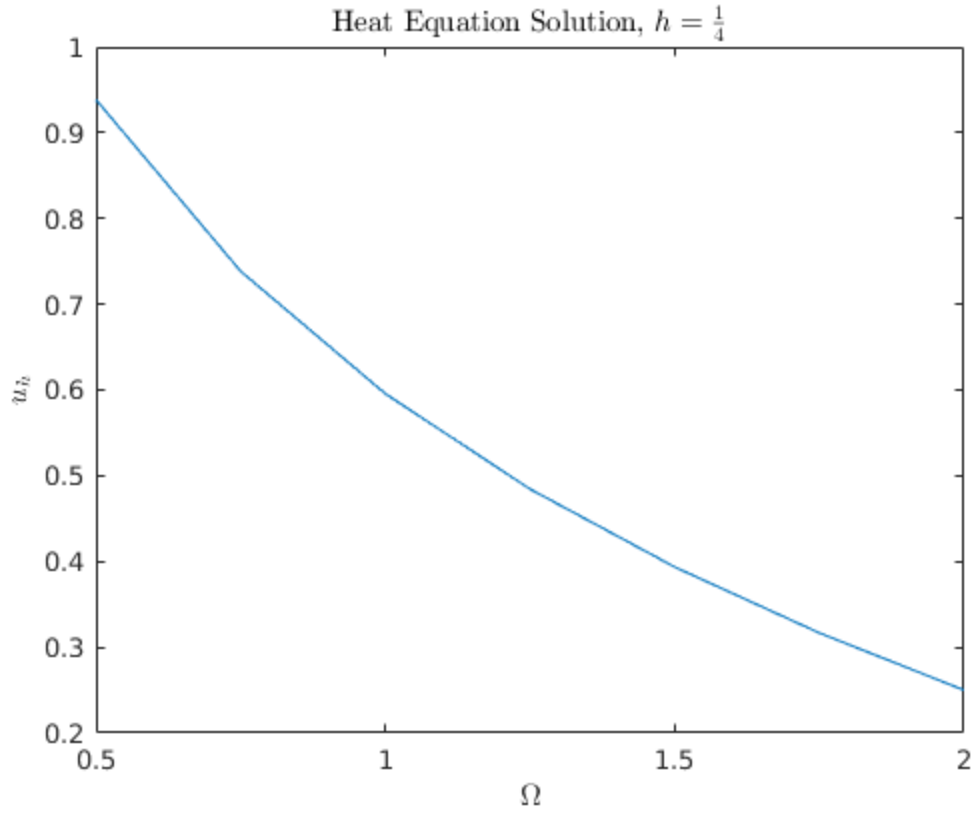
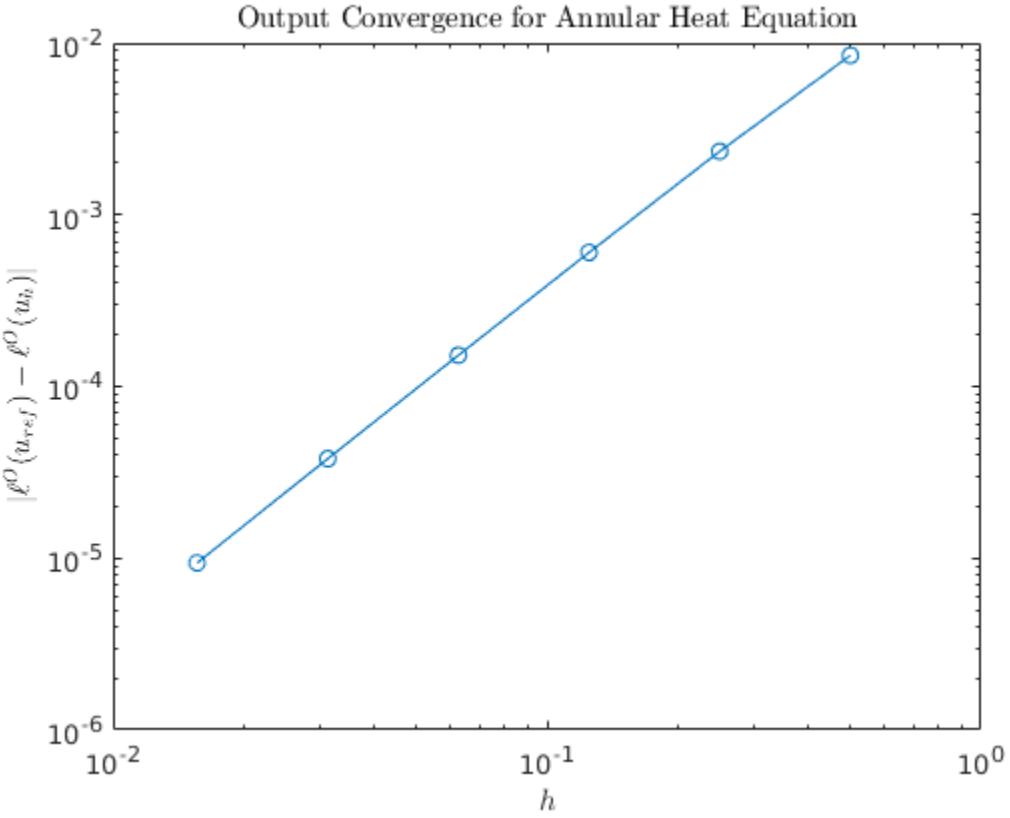
AER1418 Assignment 1 - Geoff Donoghue

January 17th 2019

```
% This .m file solves a one dimensional heat problem using a finite
 element
% method with linear polynomials and an arbitary uniform mesh spacing

% Define boundary conditions for annular radius
D.r_in = 1/2;
D.r_out = 2;

% Define problem parameters, thermal conductivity, heat flux, Biot
 Number
D.k = 1;
D.g = 1;
D.B = 1;

% Define mesh spacing parameter used to calculate reference output
h_ref = 1/512;

% Calculate reference output, used to estimate error and convergence
ref_output = solve(h_ref,D);

% Define array of mesh spacing parameters used to calculate
 convergence
h = 1./2.^(1:6);

% Solve FEM problem on each mesh, storing outputs
outputs = zeros(1, length(h));
for i = 1:length(h)
    outputs(i) = solve(h(i),D);
end

% Calaculate error in output based on previous reference value
error = abs(outputs - ref_output);

% Plot error as a function of mesh spacing parameters
figure(2)
loglog(h,error,'o-')
ylabel('$|\ell^O(u_{ref}) - \ell^O(u_{h})|$','interpreter','latex')
xlabel('$h$','interpreter','latex')
title('Output Convergence for Annular Heat
 Equation','interpreter','latex')

% Calculate convergence rate in error based on last two meshes
convrate = (log((error(end))/(error(end-1))))/(log((h(end))/
(h(end-1))));


function output = solve(h,D)
```

```
% This function builds the stiffness matrix and load vector for the
 above

% Define one-dimensional mesh, which is a simple array
x = D.r_in:h:D.r_out;
N = length(x);

% Preallocate memory for stiffness matrix and load vector
A = zeros(N,N);
f = zeros(N,1);

% Loop over nodes, filling in elements as appropriate
for i = 1:N
    % Each element is filled according to the weak form
    if x(i) == D.r_in
        A(i,i) = D.k/2/(x(i+1)-x(i))^2*(x(i+1)^2 - x(i)^2);
        A(i,i+1) = D.k/2/(x(i+1)-x(i))^2*-(x(i+1)^2 - x(i)^2);
        f(i) = D.g*D.r_in;
    elseif x(i) == D.r_out
        A(i,i) = D.k/2/(x(i)-x(i-1))^2*(x(i)^2 - x(i-1)^2) +
 D.B*D.r_out;
        A(i,i-1) = D.k/2/(x(i)-x(i-1))^2*-(x(i)^2 - x(i-1)^2);
    else
        A(i,i) = D.k/2/(x(i)-x(i-1))^2*(x(i)^2 - x(i-1)^2) ...
            + D.k/2/(x(i)-x(i-1))^2*(x(i+1)^2 - x(i)^2);
        A(i,i-1) = D.k/2/(x(i+1)-x(i))^2*-(x(i)^2 - x(i-1)^2);
        A(i,i+1) = D.k/2/(x(i+1)-x(i))^2*-(x(i+1)^2 - x(i)^2);
    end
end

% Solve system for basis coefficients approximating solution
u = A\f;

% Calculate output, which is equal to linear form at inner radius
output = D.g * D.r_in * u(1);

% Plot single solution, since basis functions are always equal to 1 at
% the nodes, we can simply plot the coefficients
if h == 1/4
    figure(1)
    plot(x,u)
    ylabel('$u_{h}$','interpreter','latex')
    xlabel('$\Omega$','interpreter','latex')
    title('Heat Equation Solution, $h=\frac{1}
{4}$','interpreter','latex')
end
end
```


Heat Equation Solution, $h = \frac{1}{4}$
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
u_h
0.5
1
1.5
2
Ω


Output Convergence for Anmular Heat Equation
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
$|\ell^O(u_{ref}) - \ell^O(u_h)|$
10^{-2}
10^{-1}
10^{0}
h

Published with MATLAB® R2016b